Philips firmware by VB6ROCOD

Let's make things better...

- Home
- Code library
- How to..
- Downloads
- Blog
- Links
- E-Boda Media player

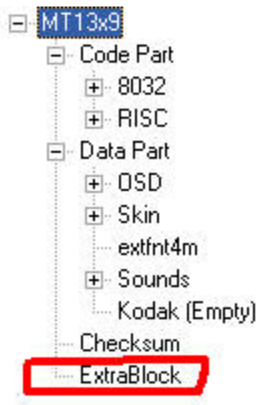
Firmware

-
- DVP5160_12
- DVP5980_12
- DVP5980S_12
- DVP3260_12
- DVP5990_12
- DVP3980_12
- DVP3160K_55
- DVP3254K_55(75)
- DVP3962_37
- DVP3980K_55(77_75)
- DVP5990_37
- DVP5990K_55(75)
- DVP5996K_51
- DVP3252_75
- DVP3350_12
- DVP3360_12
- DVP3350K_55
- DVP3360K_55
- DVP3111_58
- DVP3380_12
- DVP5992_12
- DVP3350_58
- Others...

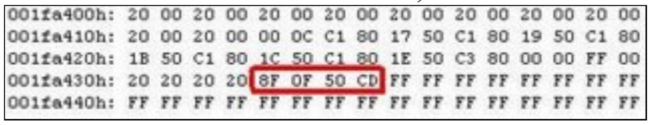
Useful Resources

- Code Library
- Upgrading Instruction
- Reflash a dead player
- ExtraBlock and MTK tools
- Info End - Non zero!
- My Tools

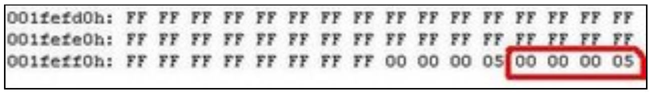
ExtraBlock and MTK Tools



New firmware from Philips, like 5990 or 3980 have at end an "ExtraBlock" of bytes. This block of bytes "fill" firmware up to 2 MB. This block is usual fill with "FF" except last 8 bytes.
If size of firmware is ~1,99MB in explorer, is a good chance for this case.
I write this article because usual tools like MTKRemaker or MTKWindows have some problems with this block.
A usual firmware end with a sequence like:
20 20 20 20 ab cd ef gh, where last 4 bytes is a calculated check sum.
A firmware with "extrablock" end with:
ab cd ef gh a1b1 c1d1 e1f1 g1h1, where last 4 bytes is a check sum over all firmware. In some firmware previous 4 bytes are used to select a "root menu".
So, in a firmware with "extra block", we have 2 check sum, normal and "over all".



```
001fa400h: 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00
001fa410h: 20 00 20 00 00 0C C1 80 17 50 C1 80 19 50 C1 80
001fa420h: 1B 50 C1 80 1C 50 C1 80 1E 50 C3 80 00 00 FF 00
001fa430h: 20 20 20 20 8F 0F 50 CD FF FF FF FF FF FF FF FF
001fa440h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```



```
001fefd0h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
001fefe0h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
001feff0h: FF FF FF FF FF FF FF FF 00 00 00 05 00 00 00 05
```

Now, return to our MTK tools. When save firmware with MTKRemaker, the only check sum calculated is first.
More than that, if extra block end with "FF", MTKRemaker "cut" extra block!!
If we save firmware with MTKWindows, the only check sum calculated is second (at end of firmware).
If you want to make change in MTKRemaker (fonts or skins)
You need an hex editor like UltraEdit or XVI32. Open firmware in editor and change last byte to 00 (or to a value <> FF). Save firmware. Now, we are sure that MTKRemaker will not cut extrablock.
Next, open firmware in MTKRemaker, make change and save. When save MTKRemaker "make" first check-sum.
For second check-sum use MTKWindows. Just open and save.
If you want to make change in MTKWindows
Open firmware in MTKWindows, make changes and save.
Next, open and save in MTKRemaker (first check-sum).
Last step, open and save in MTKWindows (last check-sum).

About check-sum in MTK firmware. This is a series of XOR. First 4 byts with next 4, result XOR with next 4 bytes and so....

For used only with MTKRemaker, I make a little program. Put this in same folder with MTKRemaker.
Open firmware.bin, program change last byte and launch MTKRemaker.
Change what you want in MTKRemaker and save firmware.bin.
After that, press button "Finish" from program.